



Estruturas de Dados e Algoritmos

Ano Letivo de 2016/17

1º Teste de Frequência

29 de março de 2017

Duração: 60 min. Com consulta restrita (formulário de PdC e fichas práticas 1 a 5)

Problema 1 (40%)

Considere uma aplicação informática que tem por objetivo registar e monitorizar dados clínicos de utentes de um estabelecimento de saúde. Nesta aplicação, existe uma ficha definida com a palavra reservada `struct` que armazena informação de um utente que sofre de hipertensão arterial: `nome` (string estilo C de tamanho 60), `idade` (int), `peso` (float), `altura` (float), `tensao_sistolica` (float) e `tensao_diastolica` (float).

- Declare a `struct`.
- Implemente a função `atualiza_tensao()` que, devolvendo `void` e aceitando como parâmetros uma ficha passada por referência, o valor atual da tensão arterial sistólica passado por cópia (float) e o valor atual da tensão arterial diastólica passado por cópia (float), atualiza os campos `tensao_sistolica` e `tensao_diastolica` da ficha passada no 1º parâmetro.
- Implemente a função `atualiza_tab()` que, aceitando como parâmetros uma tabela de fichas, uma tabela de tensões sistólicas, uma tabela de tensões diastólicas, e a dimensão das tabelas (tamanho igual para todas as tabelas), atualiza os campos `tensao_sistolica` e `tensao_diastolica` de todas as fichas da tabela passada no 1º parâmetro, usando os valores correspondentes indicados nas outras duas tabelas, e chamando obrigatoriamente a função da alínea b) para todas as fichas.

Obs.: Se não conseguir programar as funções das alíneas b) e c) respeitando integralmente as listas de parâmetros especificadas, serão consideradas outras respostas com um desconto de 50% na classificação.

Problema 2 (25%)

Considere a estrutura de dados: `struct ficha { int id; float campo_1; float campo_2; }; .`

Implemente a função recursiva `int pesquisaBinaria(ifstream &ficheiro, int inicio, int fim, float valor)` para realizar uma pesquisa binária de valor no campo `campo_1` das fichas armazenadas no `ficheiro` em modo binário. Pode assumir que o `ficheiro` já se encontra aberto e que as fichas estão ordenadas por ordem crescente segundo os valores guardados em `campo_1`. A função deve devolver a posição no `ficheiro` da primeira ficha encontrada contendo `valor` no campo `campo_1`, entre os índices `inicio` e `fim` do `ficheiro` (posições absolutas a partir do início do `ficheiro`), ou -1 se não for encontrada a ficha pretendida.

Recorde que `ficheiro.seekg(pos)` ajusta o ponteiro para leitura na posição absoluta `pos` a partir do início do `ficheiro` e que a função `ficheiro.read(unsigned char *buffer, int n)` lê em modo binário `n` bytes de informação do `ficheiro`, guardando a informação lida na tabela `buffer`.

Problema 3 (35%)

A classe de objetos `CDuracao` permite definir uma duração temporal em horas e minutos. Possui os atributos `horas` e `minutos` (tipo `int`) e os métodos seguintes: i) um construtor por defeito/omissão (objeto com uma duração por defeito igual a 1h30m); ii) um construtor por enumeração (recebe dois inteiros com uma duração qualquer em horas e minutos); iii) um construtor por cópia; iv) o método `compara()` que recebe como parâmetro outro objeto da classe `CDuracao` e devolve um n.º inteiro negativo, zero ou um n.º positivo, consoante o objeto seja menor (em duração temporal), igual ou superior ao parâmetro, respetivamente.

- Escreva a declaração da classe de objetos `CDuracao`. Nesta alínea, não deve implementar qq. método.
- Implemente os construtores por defeito, por enumeração e por cópia da classe `CDuracao`.
- Implemente o método `compara()` da classe `CDuracao`.

Nota: Em todos os problemas, não é necessário programar a função `main()`.



Estruturas de Dados e Algoritmos

Ano Letivo de 2016/17

1º Teste de Frequência

29 de março de 2017

Duração: 60 min. Com consulta restrita (formulário de PdC e fichas práticas 1 a 5)

Problema 1 (40%)

Considere uma aplicação informática que tem por objetivo registar e monitorizar dados clínicos de utentes de um estabelecimento de saúde. Nesta aplicação, existe uma ficha definida com a palavra reservada `struct` que armazena informação de um utente que sofre de diabetes: `nome` (string estilo C de tamanho 60), `mulher` (booleano que vale `true` se a pessoa é do género feminino e `false` se é do género masculino), `idade` (`int`), `peso` (`float`), `hora` (`int`) e `glicose` (`float`). O campo `hora` (`int`) serve para registar a hora (número entre 0 e 23) do dia em que foi medido o valor guardado no campo `glicose`.

- Declare a `struct`.
- Implemente a função `atualiza_glicose()` que, devolvendo `void` e aceitando como parâmetros uma ficha passada por referência, uma hora passada por cópia (`int`) e o valor da glicose medido àquela hora passado por cópia (`float`), atualiza os campos `hora` e `glicose` da ficha passada no 1º parâmetro.
- Implemente a função `atualiza_tab()` que, aceitando como parâmetros uma tabela de fichas, uma tabela de horas (`int`), uma tabela de valores de glicose (`float`), e a dimensão das tabelas (tamanho igual para todas as tabelas), atualiza os campos `hora` e `glicose` de todas as fichas da tabela passada no 1º parâmetro, usando os valores correspondentes indicados nas outras duas tabelas, e chamando obrigatoriamente a função da alínea b) para todas as fichas.

Obs.: Se não conseguir programar as funções das alíneas b) e c) respeitando integralmente as listas de parâmetros especificadas, serão consideradas outras respostas com um desconto de 50% na classificação.

Problema 2 (25%)

Considere a estrutura de dados: `struct ficha { int id; float campo_1; float campo_2; }; .`

Implemente a função recursiva `int pesquisaBinaria(ifstream &ficheiro, int inicio, int fim, float valor)` para realizar uma pesquisa binária de valor no campo `campo_2` das fichas armazenadas no `ficheiro` em modo binário. Pode assumir que o `ficheiro` já se encontra aberto e que as fichas estão ordenadas por ordem decrecente segundo os valores guardados em `campo_2`. A função deve devolver a posição no `ficheiro` da primeira ficha encontrada contendo `valor` no campo `campo_2`, entre os índices `inicio` e `fim` do `ficheiro` (posições absolutas a partir do início do `ficheiro`), ou -1 se não for encontrada a ficha pretendida.

Recorde que `ficheiro.seekg(pos)` ajusta o ponteiro para leitura na posição absoluta `pos` a partir do início do `ficheiro` e que a função `ficheiro.read(unsigned char *buffer, int n)` lê em modo binário `n` bytes de informação do `ficheiro`, guardando a informação lida na tabela `buffer`.

Problema 3 (35%)

A classe de objetos `CPeriodo` permite definir um período de tempo em minutos e segundos. Possui os atributos `minutos` e `segundos` (tipo `int`) e os métodos seguintes: i) um construtor por defeito/omissão (objeto com um período por defeito igual a 0min10s); ii) um construtor por enumeração (recebe dois inteiros com um período de tempo qualquer em minutos e segundos); iii) um construtor por cópia; iv) o método `compara()` que recebe como parâmetro outro objeto da classe `CPeriodo` e devolve um n.º inteiro negativo, zero ou um n.º positivo, consoante o objeto seja menor (em duração temporal), igual ou superior ao parâmetro, respetivamente.

- Escreva a declaração da classe de objetos `CPeriodo`. Nesta alínea, não deve implementar qq. método.
- Implemente os construtores por defeito, por enumeração e por cópia da classe `CPeriodo`.
- Implemente o método `compara()` da classe `CPeriodo`.

Nota: Em todos os problemas, não é necessário programar a função `main()`.